Class 05: Linear Classifiers

Martin Slawski

Volgenau School of Engineering
Department of Statistics

February 22nd, 2018

Linear classification

We suppose that $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$ (two classes).

As before, we work with a sample $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$ drawn i.i.d. from a probability distribution P on $\mathcal{X} \times \mathcal{Y}$.

A **linear classifier** is a map $g : \mathcal{X} \to \{-1, 1\}$ of the form

$$x \mapsto g(x) = \mathsf{sign}(f(x)),$$

where f is an element of the hypothesis class

$$\mathcal{F} = \left\{ f : \; x \mapsto f(x) = w_0 + \sum_{j=1}^{D} w_j \phi_j(x), \;\; w_0 \in \mathbb{R}, \;\; w \in \mathbb{R}^D \right\},$$

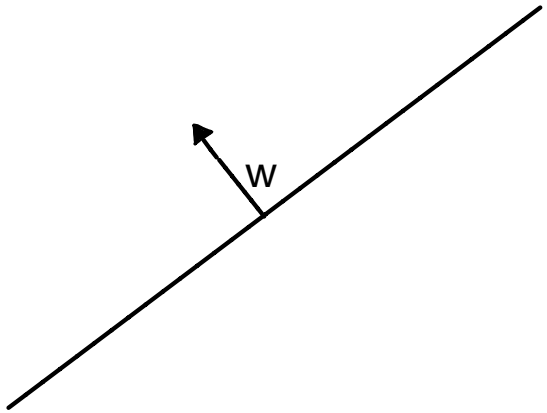
where the $\{\phi_j\}_{j=1}^D$ are fixed functions $\mathcal{X} \to \mathbb{R}$.

Linear classification

The set

$$\{x \in \mathbb{R}^D : \langle x, w \rangle + w_0 = 0\}$$

specifies a **hyperplane** in $\mathbb{R}^D$.



A linear classifier partitions the space into two halves

$$H^+ = \{x \in \mathbb{R}^D : \langle x, w \rangle + w_0 > 0\}, \quad H^- = \{x \in \mathbb{R}^D : \langle x, w \rangle + w_0 < 0\}.$$

Linear classification

As opposed to linear regression, where we can eliminate the intercept by centering the data, we do keep w_0.

If $w_0 = 0$, then the hyperplane must contain the origin. We would like to avoid this additional restriction.

Linear classification

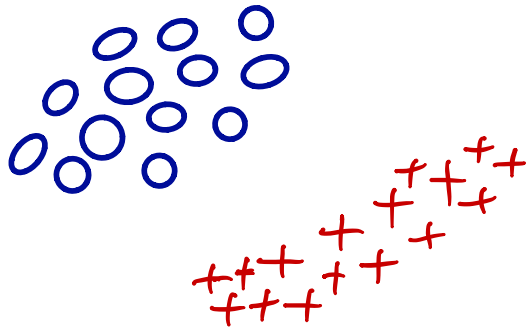
Benefits of linear classifiers:

- conceptually simple,
- computationally manageable,
- interpretable,
- not that sensitive to overfitting *

* more on this later.

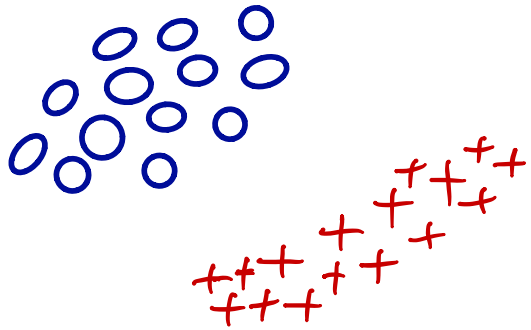
Linear classification

Two **linearly separable** classes in $\mathbb{R}^2$:

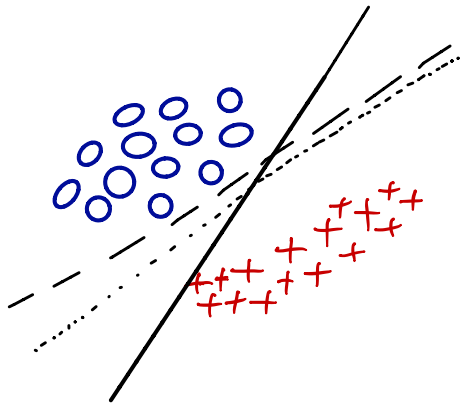


Linear classification

Two **linearly separable** classes in $\mathbb{R}^2$:



Note that there are infinitely many separating hyperplanes:



Linear classification

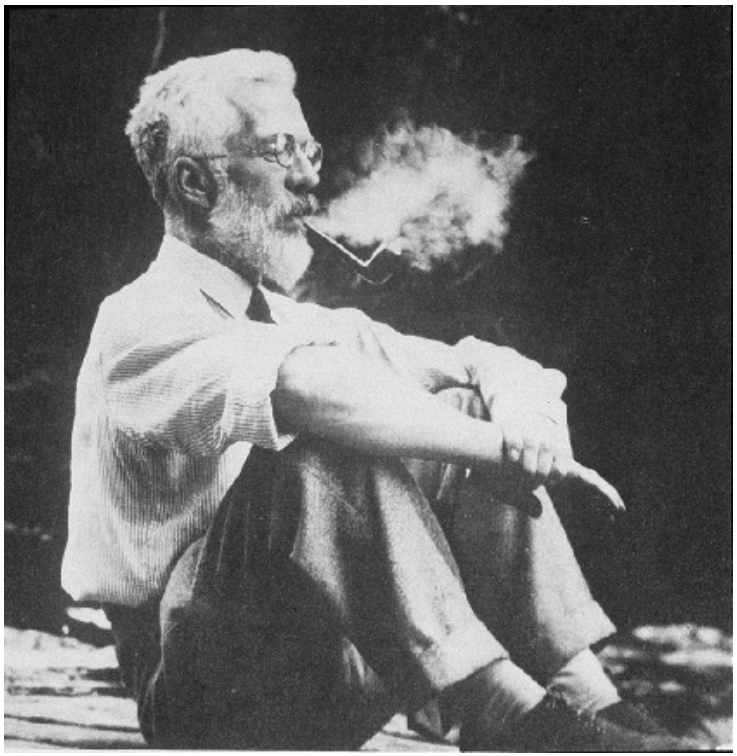
We will discuss and compare three popular approaches for constructing a hyperplane for classification:

- **L**inear **D**iscriminant **A**nalysis (LDA),
- Logistic Regression,
- Support Vector Classification.

Linear classification

LDA can be traced back to R.A. Fisher, the founder of parametric statistics and analysis of variance.

Therefore, LDA is often called Fisher's LDA.



R.A. Fisher (1890 – 1962).

Linear classification

The idea underlying LDA is **dimension reduction**:

A weight vector $w \in \mathbb{R}^D$ defines a projection into a 1-dimensional subspace.

For $x \in \mathbb{R}^D$, $\langle w, x \rangle$ can be interpreted as the coordinate of x in that subspace.

In the one-dimensional subspace, we then classify by setting a threshold t:

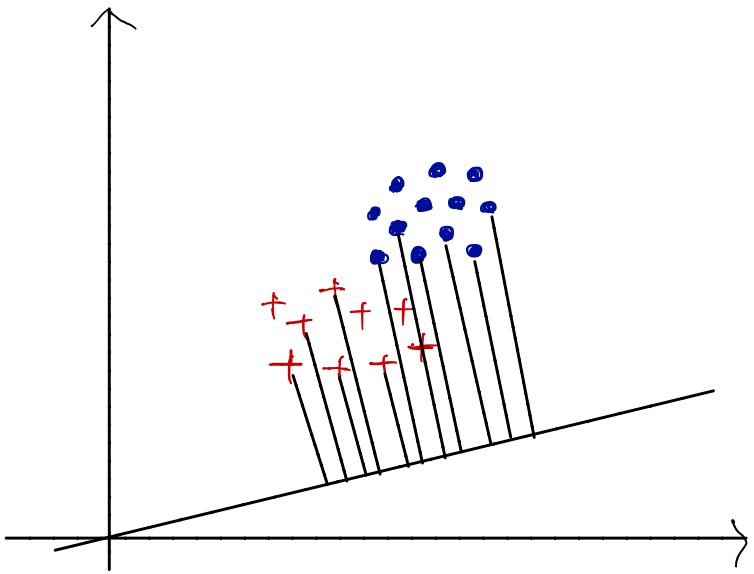
Class 1 if $\langle w, x \rangle > t$,
Class 0 otherwise.

Note that the threshold t is equivalent to w_0.

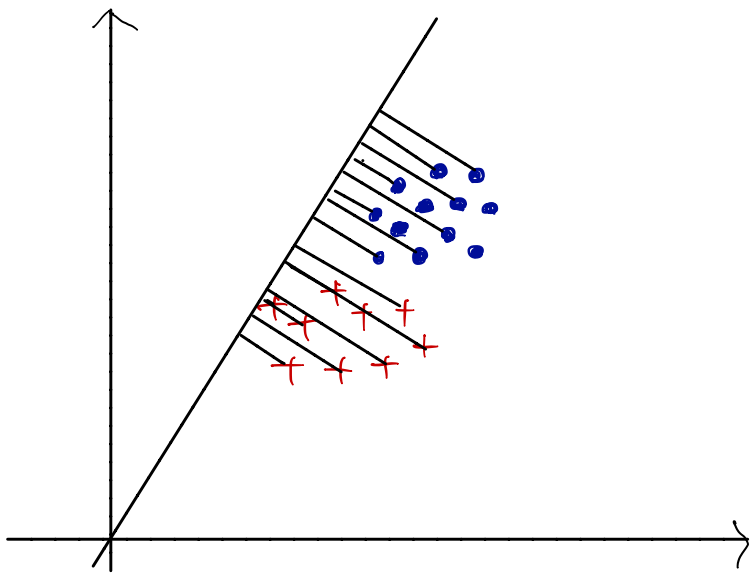
Linear classification

What are "good" projections?

bad projection



good projection



Linear classification

Fishers' criterion of an optimal projection:

Define the **class-specific means** by

$$m_{\pm} = \frac{1}{n_{\pm}} \sum_{i:\ Y_i = \pm 1} \Phi(X_i), \qquad n_{\pm} := |\{i:\ Y_i = \pm 1\}|.,$$

the **within-class scatter matrix**

$$S_W = \sum_{i:\ Y_i = 1} (\Phi(X_i) - m_+)(\Phi(X_i) - m_+)^\top + \\ + \sum_{i:\ Y_i = -1} (\Phi(X_i) - m_-)(\Phi(X_i) - m_-)^\top$$

and the **between-class scatter matrix**

$$S_B = (m_+ - m_-)(m_+ - m_-)^\top.$$

Linear classification

Fishers' criterion of an optimal projection:

$$w^* \in \operatorname*{argmax}_{w \in \mathbb{R}^D} \frac{w^\top S_B w}{w^\top S_W w} = \frac{\text{between-class scatter after projection on } w}{\text{within-class scatter after projection on } w}$$

Observe that the criterion does not depend on $\|w\|_2$.

We hence look for a direction w so that

- the projected centroids $\langle w, m_+ \rangle$ and $\langle w, m_- \rangle$ are far apart,
- the projected data are close to their respective centroid.

Linear classification

Computation of the direction w^*:

Theorem

Let us fix the sign of w^* by requiring $\langle w^*, m_+ - m_- \rangle > 0$. Then any optimal weight vector w^* of the LDA problem is proportional to a solution of the linear system of equations

$$S_W w = (m_+ - m_-).$$

In particular, if S_W is invertible, then

$$w^* \propto S_W^{-1}(m_+ - m_-)$$

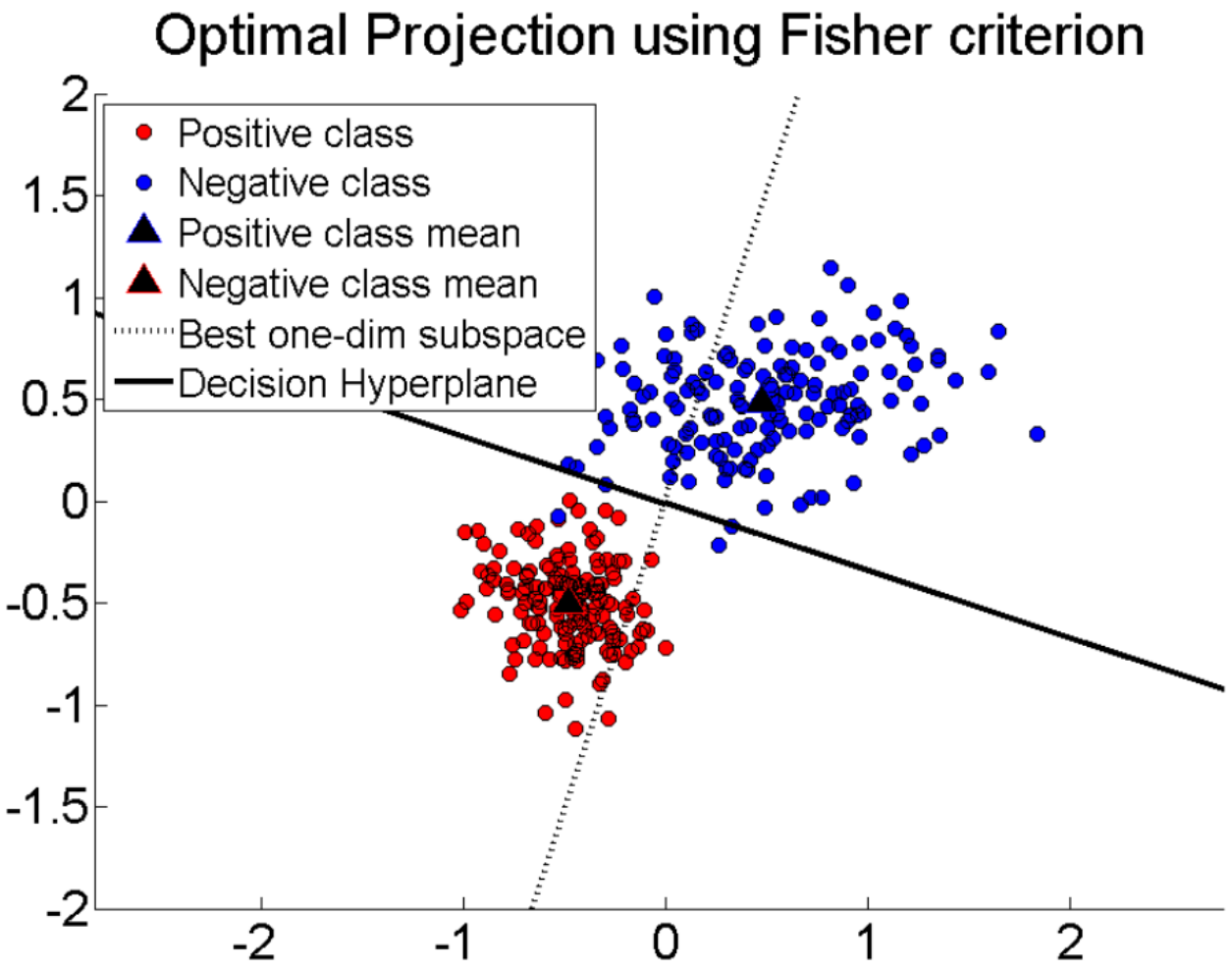
Linear classification

Given w^*, we optimize w_0 by minimizing the 0-1 loss on the training data:

$$w_0^* \in \operatorname*{argmin}_{w_0} \frac{1}{n} \sum_{i=1}^{n} I(Y_i \neq \mathsf{sign}(w_0 + \langle w^*, \Phi(X_i \rangle))$$

Note that this can be done in $O(n)$ time.

Linear classification



Linear classification

How does LDA fit into the framework of ERM?

Theorem

LDA is equivalent to empirical risk minimization with **squared loss**:

Letting $\mathbf{\Phi} = (\phi_j(X_i))_{1 \le i \le n, 1 \le j \le D}$ *denote the feature matrix and* $\mathbf{y} = (Y_i)_{i=1}^n$ *the vector of labels, consider*

$$\min_{w_0, w} \|\mathbf{y} - w_0 \mathbf{1} - \mathbf{\Phi} w\|_2^2 \quad (\#).$$

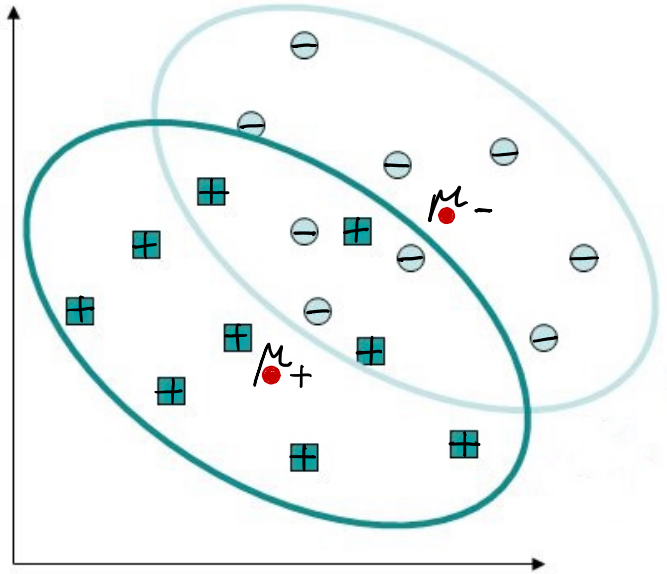
Then any optimal w *for* $(\#)$ *is proportional to an optimal solution* w^* *of the LDA problem.*

Linear classification

Consider the following generative model for the data:

$$X|Y = +1 \sim N(\mu_+, \Sigma),$$
$$X|Y = -1 \sim N(\mu_-, \Sigma),$$

$$\mathbf{P}(Y = +1) = p,$$
$$\mathbf{P}(Y = -1) = 1 - p.$$



- where $\mu_\pm \in \mathbb{R}^d$,
- $\Sigma \in \mathbb{R}^{d \times d}$ symmetric positive definite.

$\rightsquigarrow$ The class-specific distributions are multivariate Normal with identical covariance (same for both classes).

Linear classification

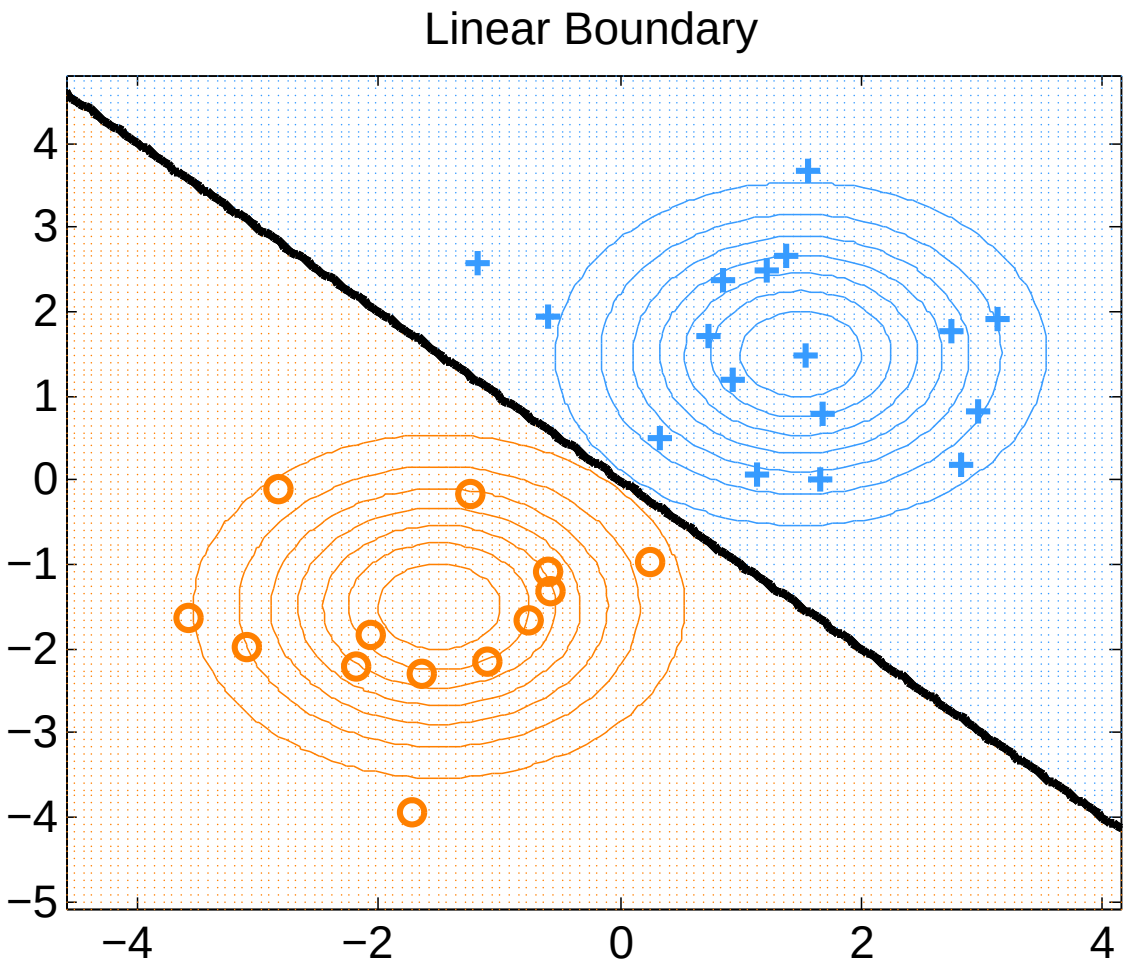
It is not hard to show that under the generative model of the previous slide, the Bayes rule is given by

$$f^*(x) = \text{sign}\Bigg(x^\top \underbrace{\Sigma^{-1}(\mu_+ - \mu_-)}_{w} + \underbrace{-\frac{1}{2}\left(\mu_+^\top \Sigma^{-1}\mu_+ - \mu_-^\top \Sigma^{-1}\mu_-\right) + \log\left(\frac{p}{1-p}\right)}_{w_0}\Bigg),$$

i.e., the Bayes rule is a linear classifier.

Linear classification

Visualization for $d = 2$:



Linear classification

Hence, if the distribution of X is a mixture of two Gaussian distributions with the same covariance, LDA "implements" the Bayes' rule.

This "optimality" result of LDA has to be interpreted with caution:

- the assumption on the data-generating model is rather restrictive
  - multivariate normality,
  - identical covariance matrices.

If the covariance matrices are different, the Bayes classifier becomes a qudratic function in general.

Linear classification

But even if the assumptions are satisfied, Σ^{-1}, μ_+, μ_- are not known and have to be estimated from $\mathcal{D}_n$:

- Estimating Σ^{-1} or even $\Sigma^{-1}(\mu_+ - \mu_-)$ is difficult if d is large relative to n (Bickel & Levina, 2004).

Linear classification

Logistic regression:

A generative model with a less restrictive assumption:

$$\mathbf{P}(Y = 1|X = x) = \frac{\exp(\overline{w}_0 + \langle \overline{w}, \Phi(x) \rangle)}{1 + \exp(\overline{w}_0 + \langle \overline{w}, \Phi(x) \rangle)}$$

$$\Leftrightarrow \quad \log \left(\frac{\mathbf{P}(Y = 1|X = x)}{\mathbf{P}(Y = -1|X = x)} \right) = \overline{w}_0 + \langle \overline{w}, \Phi(x) \rangle$$

Observe that in this case, the Bayes classfier is a linear classfier in the features $\Phi(x)$.

Linear classification

Given a sample $\mathcal{D}_n$, it is standard to estimate $\overline{w}_0, \overline{w}$ by the method of maximum likelihood.

This is equivalent to ERM with respect to the linear hypothesis class of slide 1 and the logistic loss ($\rightarrow$ Class 03 and Homework 2):

$$\min_{w_0, w} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-Y_i(w_0 + \Phi(X_i)^\top w)))$$

Since the logistic loss is a convex function of the margin $yf(x)$ and since $f(x) = w_0 + \Phi(x)^\top w$ is affine in (w_0, w), the above optimization problem is **convex**.

Linear classification

Following the principle of regularized ERM, we can add our favorite (ideally convex) regularizer to the previous optimization problem:

$$\min_{w_0, w} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-Y_i(w_0 + \Phi(X_i)^\top w))) + \lambda \Omega(w).$$

It is uncommon to use regularization for the term w_0.

The most popular choices for the regularizer are (as in linear regression):

- $\Omega(w) = \|w\|_2^2$,
- $\Omega(w) = \|w\|_1$.

Linear classification

If the training data are linearly separable, the use of a regularizer becomes important:

It can be shown that in this case – without regularizer – the optimization problem is not bounded from below.

In practice, this means that whatever routine we use for optimization, it does not converge; with increasing number iterations, we will see that $\|w\|_2 \to \infty$.

Linear classification

Independent of that, Regularization is helpful to prevent/mitigate overfitting.

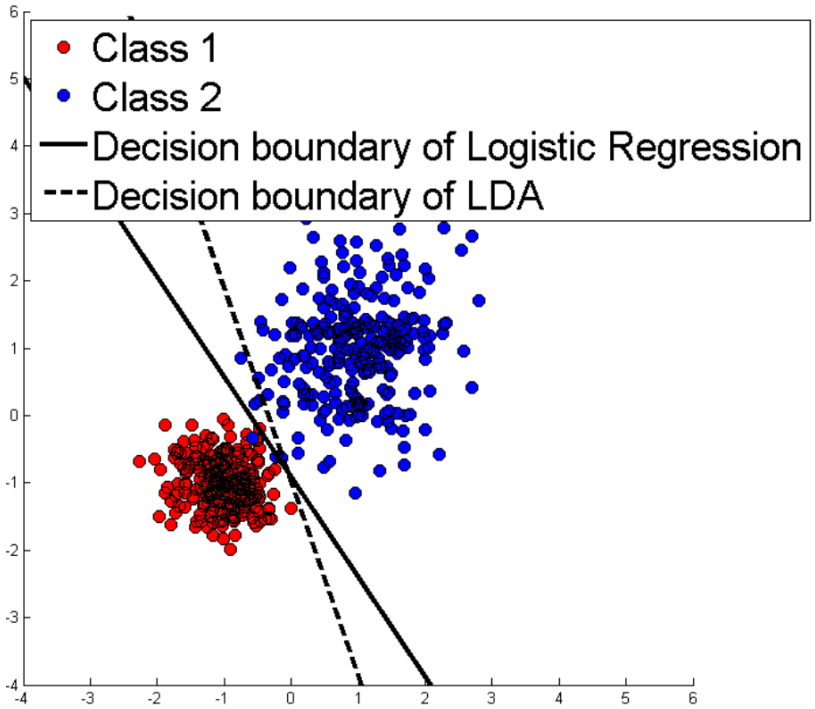
Even for a linear classifier, overfitting becomes a serious concern once D is of the same order of magnitude as n (or larger).

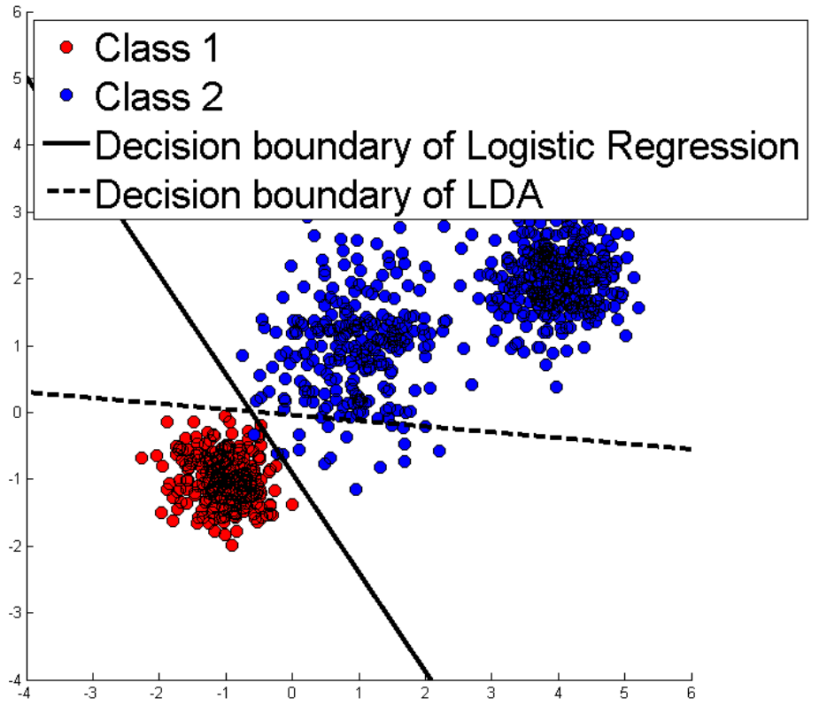
In particular, we have the following:

With a linear classifier in $\mathbb{R}^D$, it is generally possible to separate an *arbitrary* collection of up to $D+1$ samples.

Linear classification

LDA vs. logistic regression





Linear classification

The previous slides reveals a deficiency of LDA relative to logistic regression:

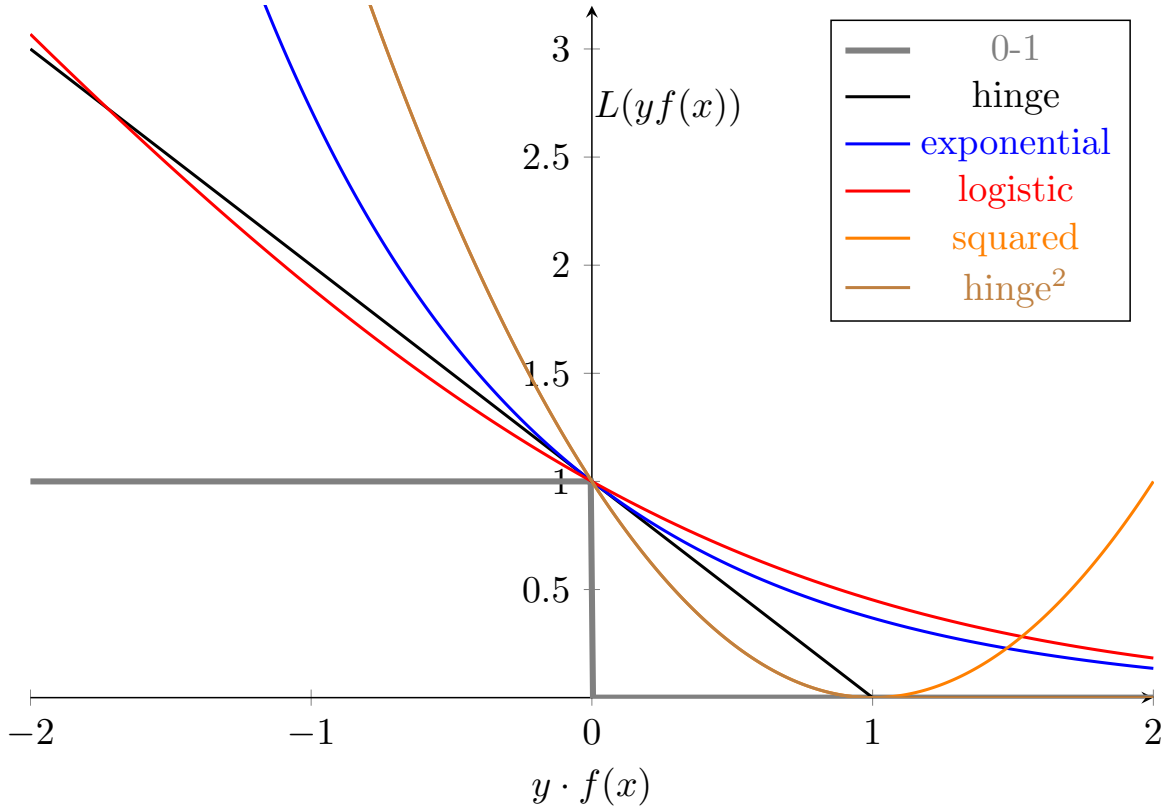
In the left hand side, both have similar decision boundaries and perform about equally well.

In the right hand side, a new "cluster" from Class 2 far away from the decision boundary has been added.

Perhaps surprisingly, this new set of points negatively affects LDA, while the decision boundary of logistic regression remains essentially unchanged.

Linear classification

The reason for the (suboptimal) behavior of LDA is the use of squared loss in a classification problem:

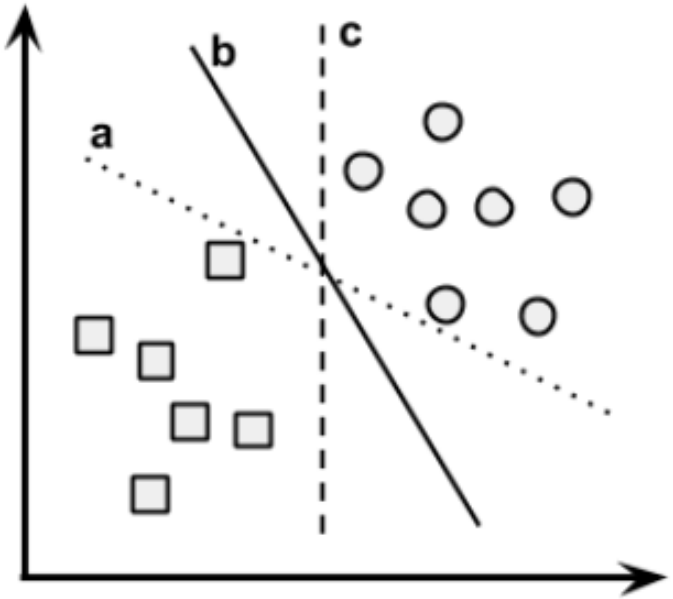


For squared loss, points with large margin are penalized!

Linear classification

Support Vector Machine (SVM)
(or support vector classification):

For a linearly separable $\mathcal{D}_n$, the **hard margin** SVM constructs a separating hyperplane with **maximum ℓ_2-margin** (to be introduced):



Linear classification

The **soft margin** SVM is a natural extension when $\mathcal{D}_n$ is not linearly separable.

While the concept of a maximum margin separating hyperplane is based on geometric considerations, it can be shown that it has a **sound statistical foundation** in the framework of regularized empirical risk minimization.

Linear classification

The key advantages of SVM over LDA and logistic regression are the following:

- SVM does not follow any generative model
- SVM achieves **data compression**:
  its solution finally depends on a (small) subset of $\mathcal{D}_n$, the so-called **support vectors**.

Linear classification

In the sequel, we discuss the various steps that lead to standard SVM formulations.

Linear classification

Given a linear classifier

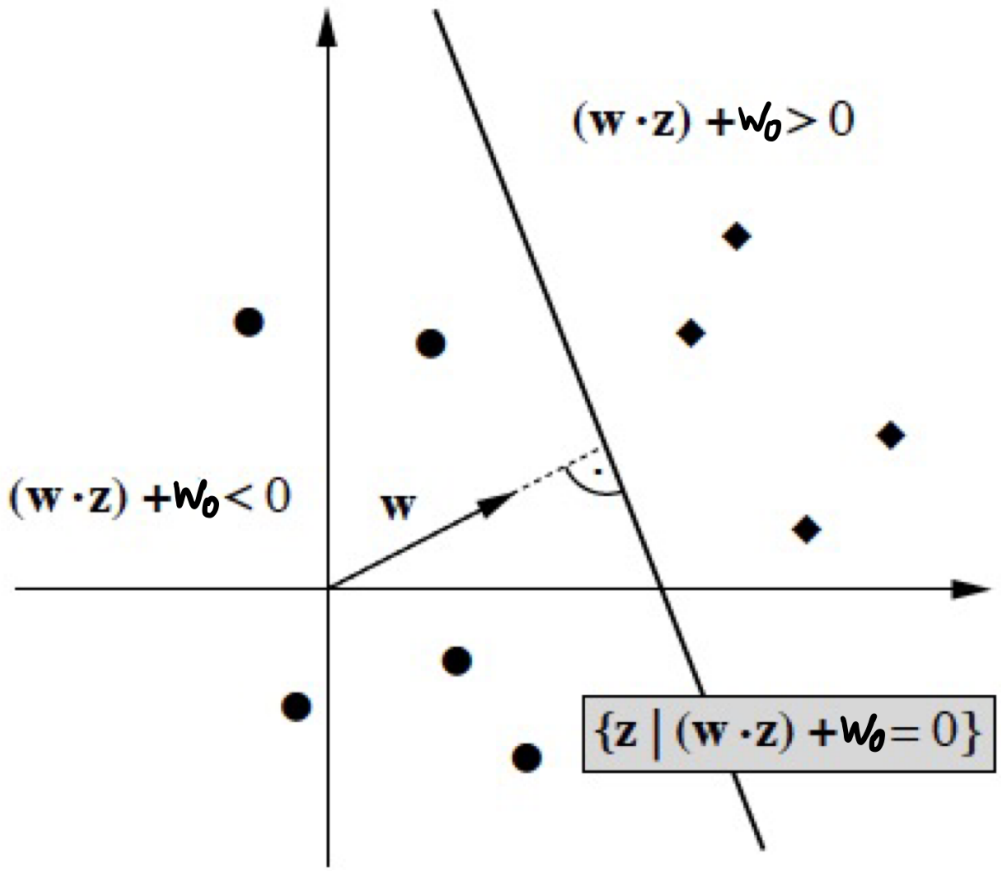
$$g(x) = \text{sign}(w_0 + \langle w, x \rangle),$$

we may re-scale w_0 and w by some constant $\gamma > 0$ which yields a new, equivalent linear classifier

$$\widetilde{g}(x) = \text{sign}(\widetilde{w}_0 + \langle \widetilde{w}, x \rangle), \quad \widetilde{w}_0 := \gamma w_0, \quad \widetilde{w} := \gamma w.$$

There are various ways of eliminating this "extra degree of freedom" (standard: require $\|w\|_2 = 1$).

Linear classification



Note:

$$\{\mathbf{z} \mid (\mathbf{w}\cdot\mathbf{z}) + w_0 = 0\}$$
$$= \{\mathbf{z} \mid (\gamma\mathbf{w}\cdot\mathbf{z}) + \gamma w_0 = 0\}$$
for $\gamma \neq 0$

Linear classification

Given a set of n points $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^D$, another way of fixing the scale of w_0 and w is by requiring that

$$\min_{1 \leq i \leq n} |w_0 + \langle w, x_i \rangle| = 1.$$

In this case, we say that the hyperplane specified by (w_0, w) is **canonical** with respect to $\{x_1, \ldots, x_n\}$.

Linear classification

Using tools from optimization, we will show in class that the ℓ_2-distance of $x \in \mathbb{R}^D$ to a hyperplane $H_{w_0,w}$ specified by (w_0, w) is given by

$$\text{dist}(x, H_{w_0,w}) := \min_{z \in H_{w_0,w}} \|z - x\|_2 = \frac{|w_0 + \langle w, x \rangle|}{\|w\|_2}.$$

Linear classification

Given a hyperplane $H_{w_0,w}$ canonical with respect to $\{x_1, \ldots, x_n\}$, i.e.,

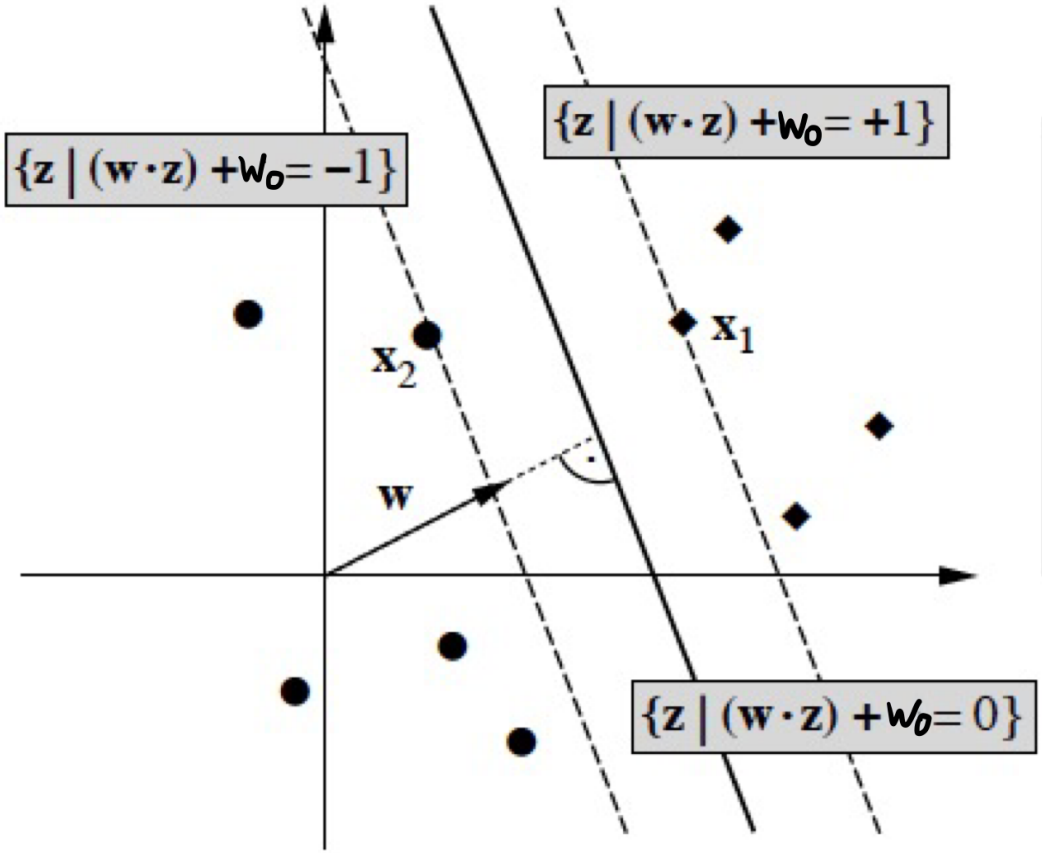
$$\min_{1 \le i \le n} |w_0 + \langle w, x_i \rangle| = 1.$$

and the fact that

$$\text{dist}(x, H_{w_0,w}) := \min_{z \in H_{w_0,w}} \|z - x\|_2 = \frac{|w_0 + \langle w, x \rangle|}{\|w\|_2},$$

we conclude that among $\{x_1, \ldots, x_n\}$, the closest point to $H_{w_0,w}$ has distance $1/\|w\|_2$ to $H_{w_0,w}$.

Linear classification



Note:

$$(\mathbf{w}\cdot\mathbf{z}_1) + = +1$$
$$(\mathbf{w}\cdot\mathbf{z}_2) + = -1$$

$$\Rightarrow \quad (\mathbf{w}\cdot(\mathbf{z}_1-\mathbf{z}_2)) = 2$$

$$\Rightarrow \left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot(\mathbf{z}_1-\mathbf{z}_2)\right) = \frac{2}{\|\mathbf{w}\|}$$

Linear classification

Given a data set $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$ that is linearly separable, i.e. there exist (w_0, w) such that

$$\text{sign}(w_0 + \langle w, \Phi(X_i)\rangle) = Y_i, \;\; i = 1, \ldots, n,$$

a **maximum margin** separating hyperplane is defined via the optimization problem

$$\max_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D} \min_{1 \le i \le n} \text{dist}(\Phi(X_i), H_{w_0,w})$$

$$\text{subject to } \text{sign}(w_0 + \langle w, \Phi(X_i)\rangle) = Y_i, \;\; i = 1, \ldots, n.$$

Linear classification

Note that if w_0, w are canonical w.r.t. $\{\Phi(X_i)\}_{i=1}^n$, the maximum margin separating hyperplane problem

$$\max_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D} \min_{1 \le i \le n} \text{dist}(\Phi(X_i), H_{w_0,w})$$

$$\text{subject to } \text{sign}(w_0 + \langle w, \Phi(X_i) \rangle) = Y_i, \;\; i = 1, \ldots, n.$$

is equivalent to

$$\max_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D} \frac{1}{\|w\|_2}$$

$$\text{subject to } |w_0 + \langle w, \Phi(X_i) \rangle| \ge 1,$$

$$\text{sign}(w_0 + \langle w, \Phi(X_i) \rangle) = Y_i, \;\; i = 1, \ldots, n.$$

Linear classification

The optimization problem

$$\max_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D} \frac{1}{\|w\|_2}$$
$$\text{subject to } |w_0 + \langle w, \Phi(X_i) \rangle| \geq 1,$$
$$\text{sign}(w_0 + \langle w, \Phi(X_i) \rangle) = Y_i, \;\; i = 1, \ldots, n.$$

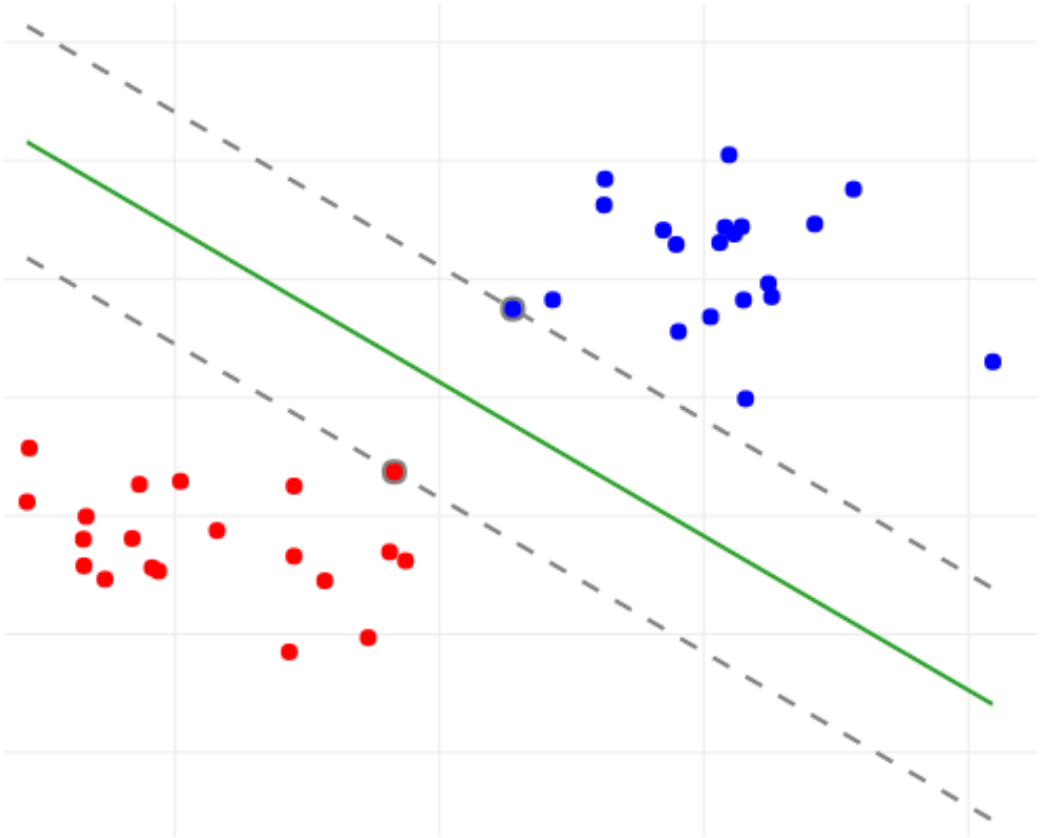
is in turn equivalent to the convex optimization problem (quadratic program)

$$\min_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D} \frac{1}{2}\|w\|_2^2$$
$$\text{subject to } Y_i(w_0 + \langle w, \Phi(X_i) \rangle) \geq 1, \;\; i = 1, \ldots, n.$$

The latter is the **hard margin SVM** problem in its standard formulation.

Linear classification

Visualization of the hard margin SVM:



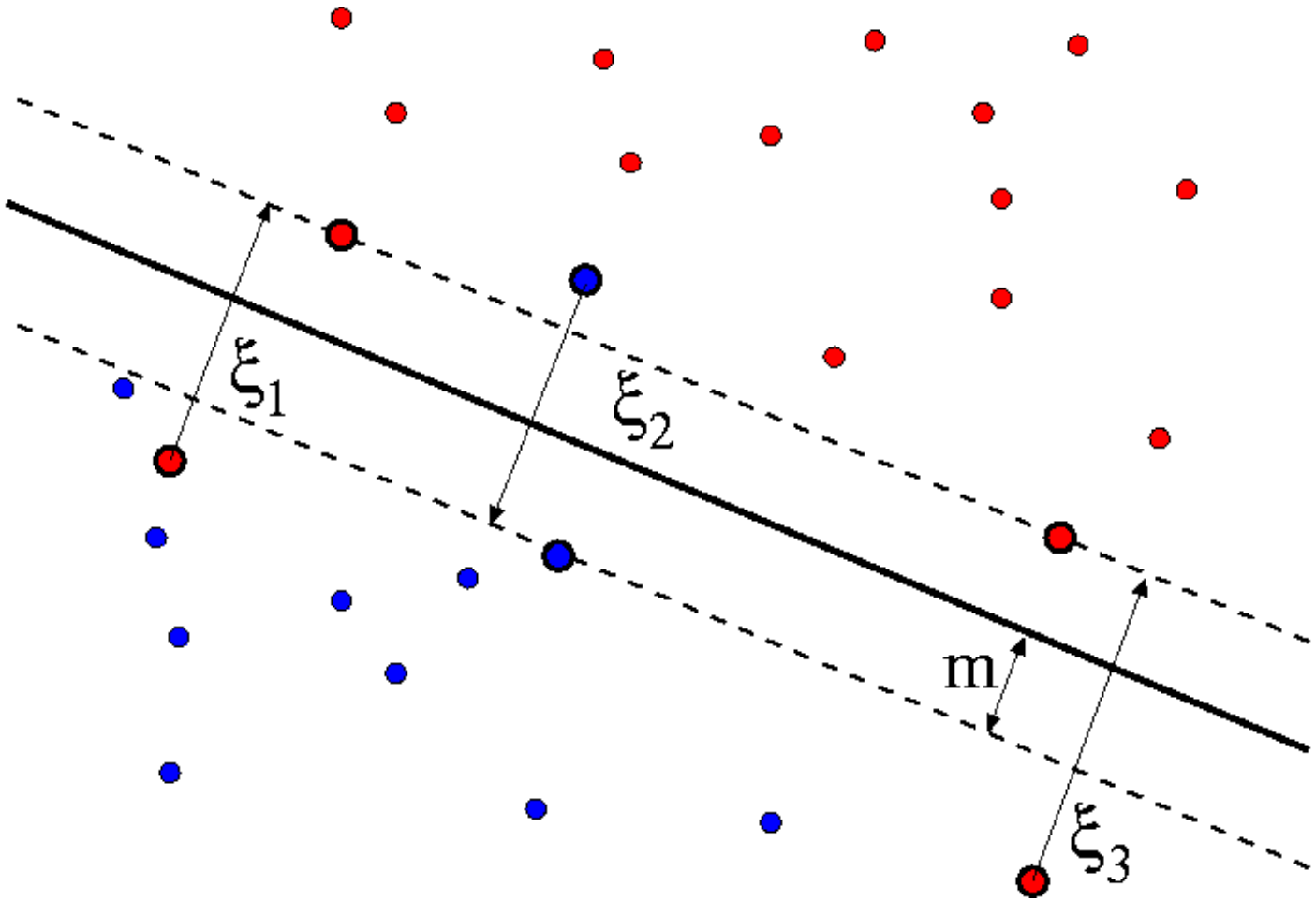
Linear classification

The hard margin SVM is feasible only if $\mathcal{D}_n$ is linearly separable. This is often not the case.

Moreover, perfect classification of $\mathcal{D}_n$ may not be desired anyway, in particular because of the danger of overfitting.

Linear classification

Solution: the **soft margin SVM**



Linear classification

After introducing "slack variables" ξ_i, $i = 1, \ldots, n$, the optimization problem of the soft margin SVM can be written as

$$\min_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D, \xi \in \mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\frac{1}{n}\sum_{i=1}^{n} \xi_i$$

$$\text{subject to } Y_i(w_0 + \langle w, \Phi(X_i)\rangle) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \;\; i = 1, \ldots, n.$$

which is still a quadratic program.

Linear classification

Soft margin SVM via the lens of **regularized ERM**:

Using that for an optimal solution (w_0^*, w^*, ξ^*) of the soft margin SVM problem, it holds that

$$\xi_i^* = \max(0, 1 - Y_i(w_0^* + \langle w^*, \Phi(X_i)\rangle)), \;\; i = 1, \ldots, n,$$

the optimization problem is equivalent to

$$\min_{w_0, w} C\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - Y_i \underbrace{(w_0 + \langle w, \Phi(X_i)\rangle)}_{f(X_i)}) + \frac{1}{2}\|w\|_2^2$$

$$\Leftrightarrow \min_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} L(Y_i f(X_i)) + \lambda\|w\|_2^2,$$

where $\mathcal{F}$ is the hypothesis class of slide 2, L is the hinge loss ($\rightarrow$ Class 03), and $\lambda = 1/(2C)$.

Linear classification

The previous slide establishes an interesting connection between

- An intuitive geometric construction,
- An "optimal" loss function for classification (tightest convex upper bound to the 0-1 loss).

Linear Classification

The optimization problem of the soft margin SVM we have looked at so far is the **primal problem**.

The corresponding **dual optimization problem** is important for the following reasons:

- it explains why the SVM achieves data compression by means of a reduction to **support vectors**
- it provides the entry point for generalizing the SVM to non-linear decision boundaries
  ($\rightarrow$ nonlinear kernel SVM, class 0?)

Linear Classification

Consider the primal soft margin SVM problem:

$$\min_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D, \xi \in \mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\frac{1}{n}\sum_{i=1}^n \xi_i$$

$$\text{subject to } Y_i(w_0 + \langle w, \Phi(X_i)\rangle) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, \;\; i = 1, \ldots, n.$$

The Lagrangian results as (blue: *non-negative* Lagrangian multipliers)

$$L(w_0, w, \xi, \alpha, \beta) = \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_{i=1}^n \xi_i$$
$$+ \sum_{i=1}^n \alpha_i \left[1 - \xi_i - Y_i\left(w_0 + \langle w, \Phi(X_i)\rangle\right)\right]$$
$$- \sum_{i=1}^n \beta_i \xi_i$$

Linear Classification

Taking the gradient of $L(w_0, w, \xi, \alpha, \beta)$ with respect to the primal variables (w_0, w, ξ) and the setting the result equal to zero, we obtain the conditions

$$w^* = \sum_{i=1}^{n} \alpha_i^* Y_i \Phi(X_i)$$

$$\sum_{i=1}^{n} \alpha_i^* Y_i = 0,$$

$$\beta_i^* = \frac{C}{n} - \alpha_i^*, \;\; i = 1, \ldots, n.$$

Since $\beta_i \geq 0$, $i = 1, \ldots, n$, we get an upper bound for the α_i's:

$$0 \leq \alpha_i \leq \frac{C}{n}, \;\; i = 1, \ldots, n.$$

Linear Classification

Substituting the relations of the previous slide back into the Lagrangian, the dual optimization problem results as

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} Y_i Y_{i'} \langle \Phi(X_i), \Phi(X_{i'}) \rangle$$

$$\text{subject to } \sum_{i=1}^{n} \alpha_i Y_i = 0$$

$$0 \leq \alpha_i \leq \frac{C}{n}, \;\; i = 1, \ldots, n.$$

Linear Classification

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} Y_i Y_{i'} \langle \Phi(X_i), \Phi(X_{i'}) \rangle$$

$$\text{subject to } \sum_{i=1}^{n} \alpha_i Y_i = 0$$

$$0 \leq \alpha_i \leq \frac{C}{n}, \;\; i = 1, \ldots, n.$$

Most of the popular implementations of the SVM are based on the dual optimization problem.

Input:

- $\mathbf{y} = (Y_i)_{i=1}^n$, $\mathbf{\Phi}\mathbf{\Phi}^\top = (\langle \Phi(X_i), \Phi(X_{i'}) \rangle)_{1 \leq i, i' \leq n}$, and C

Output:

- α_i^*, $i = 1, \ldots, n$, and w_0^*.

Linear classification

In fact, the output is typically given in the more compact form

$$\{\alpha_i^*,\ i \in \mathcal{S}^*\}$$

where

$$\mathcal{S}^* = \{i :\ \alpha_i^* > 0\}$$

is the index set of the **support vectors**. These alone determine the separating hyperplane: we have

$$w^* = \sum_{i=1}^{n} \alpha_i^* Y_i \Phi(X_i) = \sum_{i \in \mathcal{S}^*} \alpha_i^* Y_i \Phi(X_i).$$

Moreover, let $\widetilde{\mathcal{S}}^* = \{i :\ 0 < \alpha_i^* < C/n\} \subseteq \mathcal{S}^*$. Then:

$$w_0^* = 1 - \frac{1}{|\widetilde{\mathcal{S}}^*|} \sum_{i \in \widetilde{\mathcal{S}}^*} \langle w^*, \Phi(X_i) \rangle$$

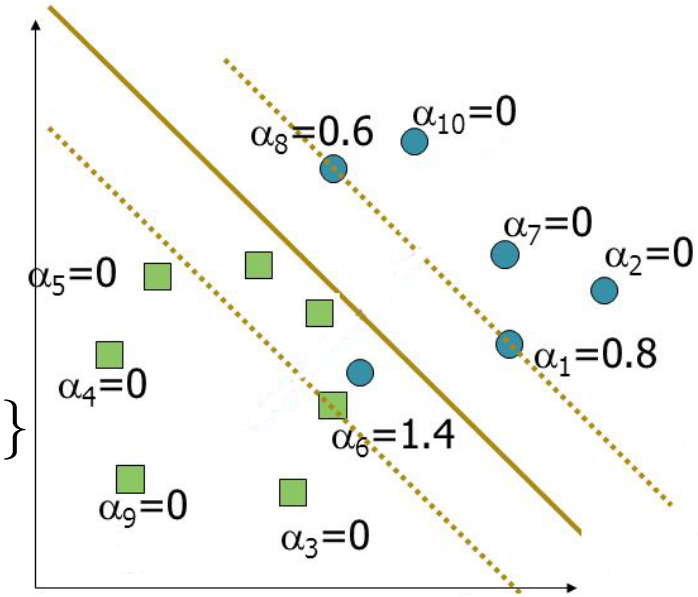
Linear classification

Geometric interpretation of support vectors:

We distinguish **four** groups of points:

1) Non-support vectors: $\{i : \alpha_i^* = 0\}$

2) Support vectors: $\mathcal{S}^* = \{i : 0 < \alpha_i^* \leq C/n\}$



2.1) Points lying on the margin: $\widetilde{\mathcal{S}}^* = \{i : 0 < \alpha_i^* < C/n\}$

2.2) Points violating the margin $\{i : \alpha_i^* = C/n\} = \{i : \xi_i^* > 0\}$

2.2.1) Points being misclassified $\{i : \xi_i^* > 1\}$

Linear classification

Linear SVM in Python: a guided tour (in two dimensions)

Function `svm.SVC` in `scikit-learn` which provides an interface to LIBSVM, a popular solver for data sets of moderate size (n in the order of at most a few thousands).

LIBSVM solves the dual formulation of the soft margin SVM.

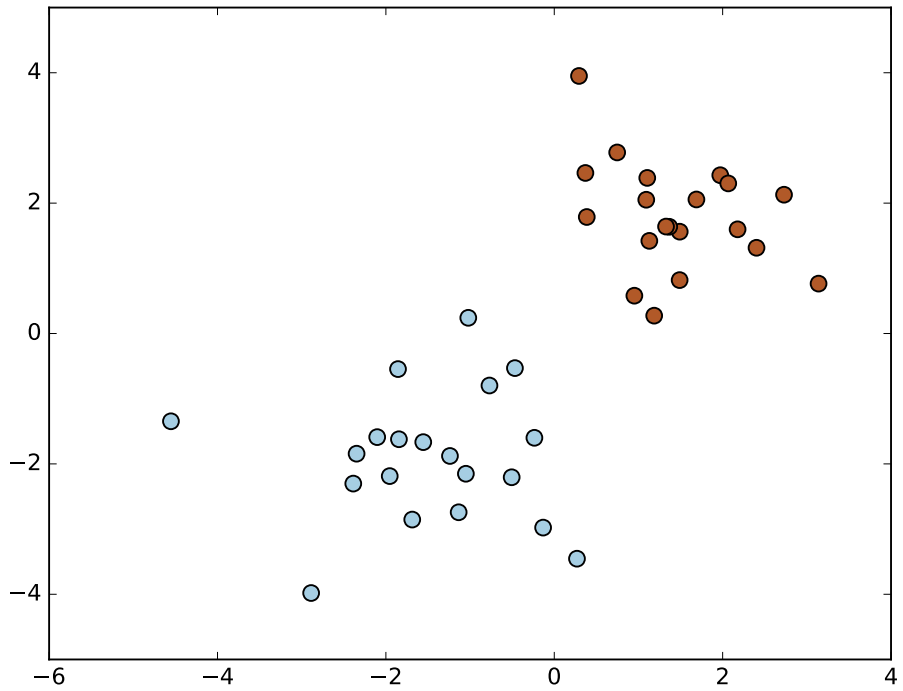
R also provides an interface to LIBSVM.

There is also a solver for the primal soft margin SVM called LIBLINEAR (with python interface `sklearn.svm.LinearSVC`)

Linear classification

A linearly separable training data set:

```
np.random.seed(0)
Y = [-1] * 20 + [1] * 20
X = np.r_[np.random.randn(20,2) - [2,2], np.random.randn(20,2) + [2,2]]
```



Linear classification

Function svm.SVC in scikit-learn:

Important arguments:

- X,Y: input data
- C: the constant in front of the objective $\sum_{i=1}^{n} \xi_i$
  (note that I used C/n instead of C)
- kernel: so far, we set this to "linear". Other choices to be discussed later
  ($\rightarrow$ kernel-based learning algorithms)

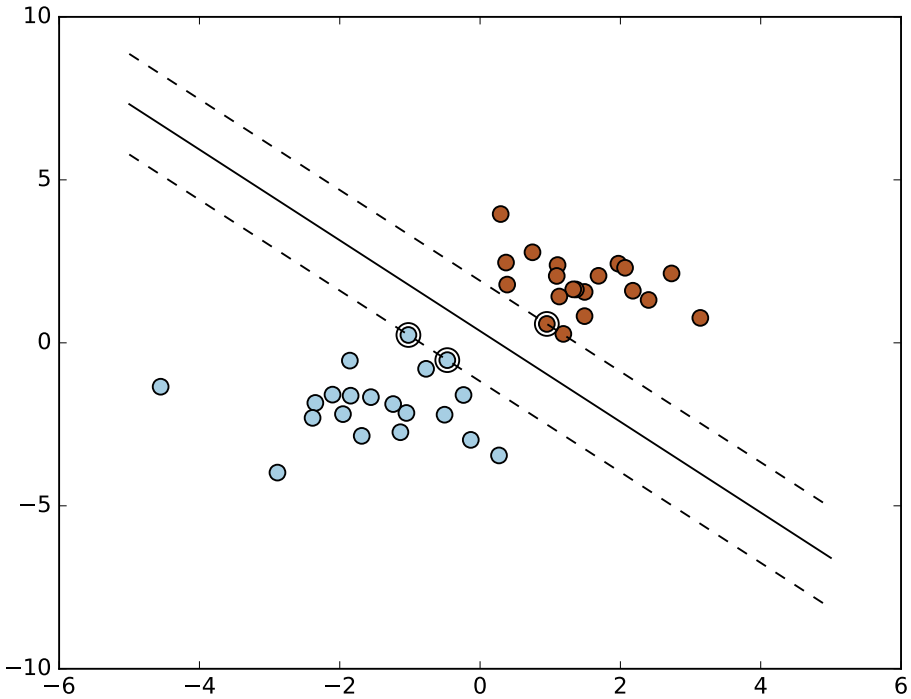
Important output:

- `coef_, intercept_`: optimal primal variables w^* and w_0^*
- `support_`: index set of support vectors $\mathcal{S}^*$
- `support_vectors`: $\{X_i, i \in \mathcal{S}^*\}$
- `dual_coef_`: $\{\alpha_i^* Y_i\}_{i \in \mathcal{S}^*}$ (note the scaling by Y_i)

Linear classification

Fit **hard margin** SVM ($C \to \infty$):

```
hardmargin = svm.SVC(C=1e6, kernel='linear')
hardmargin.fit(X, Y)
```



Linear classification

Accessing the support vectors and the dual coefficients $\{\alpha_i^* y_i\}_{i \in \mathcal{S}^*}$:

```
>>> hardmargin.dual_coef_
array([[-0.04825885, -0.56891844,  0.61717729]])
```

Computing w^* from the dual coefficients as $w^* = \sum_{i \in \mathcal{S}^*} (\alpha_i y_i) X_i$:

```
wcheck = np.dot(hardmargin.dual_coef_, X[svs_indices,:])
```

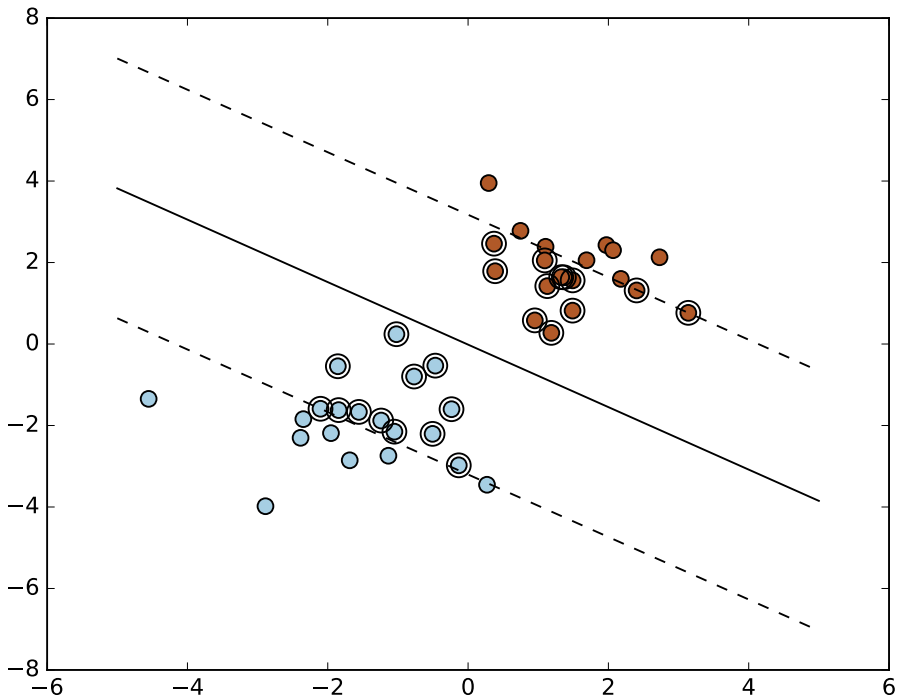
It is the same as what as returned in attribute `coef`:

```
>>> np.linalg.norm(hardmargin.coef_ - wcheck)
0.0
```

Linear classification

Set $C = 0.01$:

```
softmargin = svm.SVC(C=1e-2, kernel='linear')
softmargin.fit(X, Y)
```



Linear classification

Now there are plenty of support vectors:

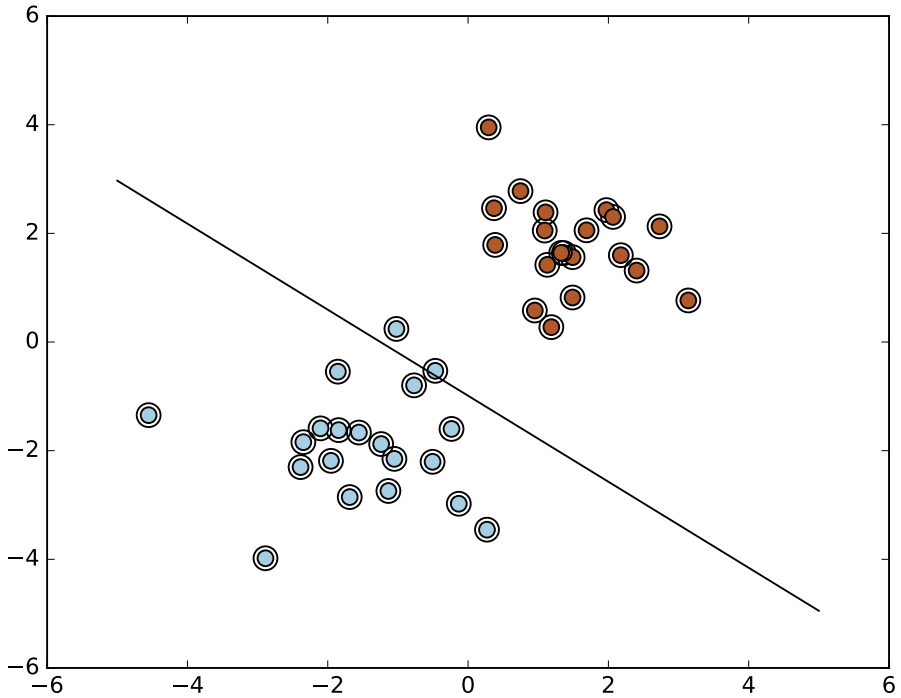
```
>>> softmargin.dual_coef_[0]
array([[-0.01, -0.01, -0.01, ..., 0.000363, 0.01, 0.01]])
>> len(softmargin.dual_coef_[0])
24
```

Note that most of the α_i^* are exactly equal to $C = 0.01$ (i.e. they lie inside the margin).

Linear classification

Decreasing C to 0.001:

```
softmargin = svm.SVC(C=1e-3, kernel='linear')
softmargin.fit(X, Y)
```



Linear classification

Some common variants of the soft margin SVM:

1) The ℓ_1-norm soft margin SVM:

$$\min_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D, \xi \in \mathbb{R}^n} \|w\|_1 + C\frac{1}{n}\sum_{i=1}^{n} \xi_i$$

$$\text{subject to } Y_i(w_0 + \langle w, \Phi(X_i)\rangle) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \;\; i = 1, \ldots, n.$$

Performs variable selection according to the same heuristic as used in the LASSO.

The geometric interpretation becomes different.

Linear classification

Some common variants of the soft margin SVM:

2) Squared hinge loss soft margin SVM:

$$\min_{w_0 \in \mathbb{R}, w \in \mathbb{R}^D, \xi \in \mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\frac{1}{n}\sum_{i=1}^{n} \xi_i^2$$

$$\text{subject to } Y_i(w_0 + \langle w, \Phi(X_i)\rangle) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \;\; i = 1, \ldots, n.$$

Linear classification

The inventors of the SVM (awarded the Paris Kanellakis Award 2008):

Corinna Cortes
Google Research, NY

Vladimir Vapnik
Facebook AI Research, NY